AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for automatically ~~computing~~ generating code to compute a derivative of a numerical expression within a digital computer system, comprising:

receiving a representation of the numerical expression within the digital computer system, wherein the numerical expression includes one or more independent variables;

storing the representation in a computer memory;

forming an expression tree for the derivative of the numerical expression with respect to an independent variable, wherein the expression tree makes use of temporary variables to form results of sub-expressions for computing the derivative of the numerical expression; and

wherein forming the expression tree involves seeking to introduce only temporary variables and associated sub-expressions as necessary to eliminate repeated common sub-expressions, thereby substantially minimizing the number of temporary variables; and

using the expression tree to generate code to compute the derivative of the numerical expression during a computation.

2. (Original) The method of claim 1, wherein for each temporary variable in the expression tree, the method forms an expression for a partial derivative of the temporary variable with respect to each independent variable.

3. (Original) The method of claim 2, wherein forming the expression for a given partial derivative involves defining new temporary variables and corresponding sub-expressions and new partial derivatives.

4. (Original) The method of claim 1, wherein a given sub-expression can include a multinomial that contains more than one binary operation.

5. (Original) The method of claim 1, further comprising pruning sub-expressions and associated temporary variables that are not used by the computation.

6. (Original) The method of claim 5, wherein pruning sub-expressions involves:

marking all temporary variables used in evaluating the expression tree; and

deleting unmarked temporary variables and associated sub-expressions.

7. (Original) The method of claim 1,

wherein the method is performed within a compiler; and

wherein the representation of the numerical expression is in the form of an Abstract Syntax Tree (AST).

8. (Original) The method of claim 1,

wherein the method is performed within a pre-compiler;

wherein the expression tree is in the form of a source code list; and

wherein each temporary variable and associated sub-expression in the expression tree is represented by computer code that sets the temporary variable equal to the associated sub-expression.

9. (Original) The method of claim 1, wherein the method is performed by code within a code library.

10. (Original) The method of claim 1,

wherein the computation involves interval arithmetic; and

wherein the one or more independent variables are interval variables.

11. (Currently amended) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for automatically ~~computing~~ generating code to compute a derivative of a numerical expression within a digital computer system, the method comprising:

receiving a representation of the numerical expression within the digital computer system, wherein the numerical expression includes one or more independent variables;

storing the representation in a computer memory;

forming an expression tree for the derivative of the numerical expression with respect to an independent variable, wherein the expression tree makes use of temporary variables to form results of sub-expressions for computing the derivative of the numerical expression; and

wherein forming the expression tree involves seeking to introduce only temporary variables and associated sub-expressions as necessary to eliminate repeated common sub-expressions, thereby substantially minimizing the number of temporary variables; and

using the expression tree to generate code to compute the derivative of the numerical expression during a computation.

12. (Original) The computer-readable storage medium of claim 11, wherein for each temporary variable in the expression tree, the method forms an

expression for a partial derivative of the temporary variable with respect to each independent variable.

13. (Original) The computer-readable storage medium of claim 12, wherein forming the expression for a given partial derivative involves defining new temporary variables and corresponding sub-expressions and new partial derivatives.

14. (Original) The computer-readable storage medium of claim 11, wherein a given sub-expression can include a multinomial that contains more than one binary operation.

15. (Original) The computer-readable storage medium of claim 11, wherein the method further comprises pruning sub-expressions and associated temporary variables that are not used by the computation.

16. (Original) The computer-readable storage medium of claim 15, wherein pruning sub-expressions involves:

marking all temporary variables used in evaluating the expression tree; and

deleting unmarked temporary variables and associated sub-expressions.

17. (Original) The computer-readable storage medium of claim 11,

wherein the method is performed within a compiler; and

wherein the representation of the numerical expression is in the form of an Abstract Syntax Tree (AST).

18. (Original) The computer-readable storage medium of claim 11,

wherein the method is performed within a pre-compiler;

wherein the expression tree is in the form of a source code list; and

wherein each temporary variable and associated sub-expression in the expression tree is represented by computer code that sets the temporary variable equal to the associated sub-expression.

19. (Original) The computer-readable storage medium of claim 11, wherein the method is performed by code within a code library.

20. (Original) The computer-readable storage medium of claim 11,

wherein the computation involves interval arithmetic; and

wherein the one or more independent variables are interval variables.

21. (Currently amended) An apparatus for automatically ~~computing~~ generating code to compute a derivative of a numerical expression within a digital computer system, comprising:

a receiving mechanism that is configured to receive a representation of the numerical expression within the digital computer system, wherein the numerical expression includes one or more independent variables;

a storing mechanism that is configured to store the representation in a computer memory;

an expression tree forming mechanism that is configured to form an expression tree for the derivative of the numerical expression with respect to an independent variable, wherein the expression tree makes use of temporary variables to form results of sub-expressions for computing the derivative of the numerical expression; and

wherein the expression tree forming mechanism seeks to introduce only temporary variables and associated sub-expressions as necessary to eliminate repeated common sub-expressions, thereby substantially minimizing the number of temporary variables; and

an execution mechanism that is configured to use the expression tree to generate code to compute the derivative of the numerical expression during a computation.

22. (Original) The apparatus of claim 21, wherein for each temporary variable in the expression tree, the expression tree forming mechanism is configured to form an expression for a partial derivative of the temporary variable with respect to each independent variable.

23. (Original) The apparatus of claim 22, wherein in forming the expression for a given partial derivative, the expression tree forming mechanism is configured to define new temporary variables and corresponding sub-expressions and new partial derivatives.

24. (Original) The apparatus of claim 21, wherein a given sub-expression can include a multinomial that contains more than one binary operation.

25. (Original) The apparatus of claim 21, further comprising a pruning mechanism that is configured to prune sub-expressions and associated temporary variables that are not used by the computation.

26. (Original) The apparatus of claim 25, wherein the pruning mechanism is configured to:

mark all temporary variables used in evaluating the expression tree; and to

delete unmarked temporary variables and associated sub-expressions.

27. (Original) The apparatus of claim 21,

wherein the apparatus resides within a compiler; and

wherein the representation of the numerical expression is in the form of an Abstract Syntax Tree (AST).

28. (Original) The apparatus of claim 21,

wherein the apparatus resides within a pre-compiler;

wherein the expression tree is in the form of a source code list; and

wherein each temporary variable and associated sub-expression in the expression tree is represented by computer code that sets the temporary variable equal to the associated sub-expression.

29. (Original) The apparatus of claim 21, wherein the apparatus includes code within a code library.

30. (Original) The apparatus of claim 21,

wherein the computation involves interval arithmetic; and

wherein the one or more independent variables are interval variables.